COMPUTER-ASSISTED INSTRUCTION

A SURVEY OF THE LITERATURE

January 1966

Albert E. Hickey

John M. Newton

Prepared under
Office of Naval Research
Contract Nonr 4757(00)
Project NR 154-254

ENTELEK Incorporated

Newburyport, Massachusetts, 01950

(617) 465-3000

PREFACE

This report is the first in a series to be published at six-month intervals describing the literature of computer-assisted instruction (CAI) as it has been compiled by ENTELEK under contract Nonr-4757(00). This report covers approximately one hundred documents abstracted prior to December 31, 1965. Obviously, the CAI literature is more extensive and it is hoped that subsequent reviews will be more nearly representative of the whole.

The contents of the report are organized according to the key word organization of the ENTELEK CAI information exchange. Citations given are the serial numbers assigned to documents in the ENTELEK system.

The section of this report dealing with interface equipment was written by John M. Newton. Albert E. Hickey is responsible for the balance of the report. The authors are indebted to Drs. Glenn Bryan and James Regan of the Personnel and Training Branch, Office of Naval Research, who have constructively monitored the general development of the system from its inception.

CONTENTS

I. APPLICATIONS

A. SUBJECT MATTER

Appendix A lists 41 references to CAI programs in mathematics and statistics, history, languages, science, economics, accounting, engineering, computer technology, reading, games, and industrial and military skills. These 41 citations, however, describe about 20 distinct programs, few of which are more than three study-hours in duration.

Potential application of automated instruction to other subjects is also discussed in the literature. For example, in an experiment on the discrimination learning of smoothness of machined metal surfaces, Mager and Clark (63-901) demonstrated the possibility of teaching non-objective responses, including judgements of social or nonsocial stimuli, value, and attitudes.

B. OTHER APPLICATIONS

Automated instructional systems have also been used for purposes other than instruction. Pines (65-502) reports that the Edison Responsive Environment has demonstrated therapeutic value for autistic children.

The Control Systems Laboratory at the University of Illinois has investigated a textbook tester that will permit an ordinary textbook to be entered into the PLATO system and a complete record kept of the student's work studying it (64-1202). Analysis of students' use of the text will indicate necessary text revisions and these can be made and tried out immediately.

Feurzeig (64-804) believes the Socratic System has potential value in areas other than teaching, e.g., planning, testing, interviewing, and monitoring. Stolurow (64-1101) has reported on an adaptive retrieval system for use with abstracts. Retrieval strategies adopted by the system are dependent on the user's response to abstracts.

II. SYSTEMS

A. SYSTEMS ANALYSIS

Systems analysis refers to the construction of models and procedures to optimize some function of the variables involved in the model. The function may be learning.

Stolurow (65-702) believes a general cybernetic model of an instructional system has seven components: input (display to the learner), learner, output (learner response), performance standards, comparator (comparison of output with standard), library and feedback. He has outlined the variables associated with each component.

SOCRATES is an operational model of the instructional process as seen by Stolurow (65-702, 64-901). The model is ideomorphic in that it considers the learner's characteristics as assessed before instruction and is also adaptive during instruction.

Mayer (64-902) views the trainee and instructor as operators within an information system and proposes a model for use in the application of human engineering principles and techniques to the design of instructional systems.

Egbert and Cogswell (64-202) examine a plan for a Continuous Progress School (CPS) being developed at the Brigham Young University Laboratory School and use the data for systems analysis and simulation to construct an explicit descriptive model of a future CPS. In another report (64-303) the same authors outline the use of systems analysis and computer simulation in investigating the effects of innovations and adjustments in school curricula.

B. COMMON SYSTEM CHARACTERISTICS

1. Time-Sharing

To be economical, large computer-assisted instructional systems must service many students at once. Thus, time-sharing is an important requirement.

Approved For Release 2003/04/29 : CIA-RDP84-00780R001300060035-6

"A computer-based teaching system has a number of student stations tied to one central, general purpose, high-speed computer. With such a time-sharing system, the decision-making and computational power of the central processor can be harnessed to provide for sophisticated automatic teaching techniques (Braunfeld, 62-802)." Due to the speed of the computer, a given student is seldom conscious of the fact that other persons are being serviced by the system.

Lochner (65-903) describes time-sharing in the Dartmouth College system. Equipment consists of a GE-235, a Datanet-30 Communications Controller, and a dual-access disc file. The D-30 is the master computer and schedules all nontrivial operations in the 235. In addition, the D-30 handles input/output from and to the teletypes, allocates and releases disc space, and analyses and responds to user commands. Lochner includes ideas on optimizing the use of a time-shared system.

Feurzeig (64-601) sees time-sharing of the computer as the way to make the BBN Socratic System, with its extensive vocabulary and logic requirements, economically feasible. In other papers (64-804, 64-803) he describes the use of time-sharing in the Socratic System to evaluate "collaborative time-sharing." An experiment was run in which six subjects, sitting at separate consoles, worked together on a problem. They did somewhat better through collaboration than each would likely have done individually.

In 1963 the PLATO I system at the University of Illinois Coordinated Science Laboratory was replaced by PLATO II which was used to study problems of time-sharing. Now PLATO III, based on a CDC 1604, is capable of controlling 1000 student stations, although 20 was set as an initial goal (64-1202).

In a manual (64-001), IBM explains that, with proper equipment, the IBM 1401 or 1440 operating system used for computer-assisted instruction may be time-shared so that it can simultaneously teach up to five courses, using up to 12 typewriter stations.

In Russia, the Radon complex at the Academy of Sciences of the Ukrane uses the general purpose UMShN computer with up to 25 student terminals (64-201). Expansion of the time-sharing system will allow it to teach 120 to 150 students as many as four different subjects simultaneously.

An advanced application of time-sharing is planned in a coaxial community cable network that will connect 10,000 homes in Columbia City, Maryland, to a community digital computer (65-603). The system, which will enable the near simultaneous use of the computer by 5000 people, is being designed by the Brooks Foundation in conjunction with the Maryland State Dept. of Education.

2. Partial Answer Processing

Preliminary data collected by Wodtke and his associates (65-902) indicate the importance of partial answer processing, by which the computer in a CAI system is instructed to ignore trivial errors and differences in student-response formats. Student controlled pauses in the instructional output of the system also seem important. The study was conducted at Pennsylvania State University using remote terminals of the IBM Yorktown system. The Socratic System also provides for partial answer processing through the use of a subroutine called CODEX.

C. OPERATIONAL SYSTEMS

In the past three or four years computer-assisted instruction systems have grown, more or less independently, in several laboratories. These pioneering examples of CAI share some common features, such as time-sharing just described. Each also has unique features. In this section we list and describe several well-established systems.

In the recent past, each of the following systems was restricted to the laboratory or campus on which it was generated. Recently, however, remote terminals for several of these systems have been installed at other laboratories and schools, sometimes at the opposite end of the country and even overseas. Thus, some systems can no longer be identified solely with a particular institution, except historically or administratively, but are nationwide in extent.

1. IBM Yorktown.

An extensive system is based on the IBM Watson Research Laboratories at Yorktown, New York. This system was originated with an IBM 650 RAMAC with a disc file for 6 million digits and 0.8 secs. maximum access time (64-302). I/O capability was provided by 20 Modified Inquiry Stations Type 838. In 1964 an IBM 1440 was substituted for the 650 (65-902, 65-803, 64-001).

Students at Pennsylvania State and Florida State Universities now take courses transmitted from IBM's Yorktown complex (65-902). The course is presented at a modified electric typewriter and by random-access slide projector and tape recorder.

In 1965 the IBM Yorktown system was linked by telephone wire to terminals at the Americana Hotel in New York City for the American Management Association conference on educational technology.

2. IBM Poughkeepsie

A system similar to the IBM Yorktown system is based on an IBM 1440 at Poughkeepsie. Through remote terminals located at IBM offices in Philadelphia, Los Angeles, San Francisco and Washington, D.C., IBM customer engineers will receive training while on call at their offices (65-1002, 65-910, 65-803).

3. University of California at Irvine (UCI)

Under a joint research agreement with IBM, UCI will become a computer laboratory for investigating all the ways in which the computer can aid educational institutions (65-910).

4. Dartmouth Time-Sharing System

The Dartmouth system is presently more suited to the laboratory type of instruction and handles jobs from 30 remote consoles. Equipment consists of a GE-235, a Datanet-30 Communications Controller, and a dual-access disc file. The D-30 is the master computer and schedules all non-trivial operations in the 235. The D-30 can communicate with the 235 or the dual-access file. In addition, the D-30 handles I/O from and to the teletypes, allocates and releases disc space and analyzes and responds to user commands. The 235 is used for editing, compilation, program runs, and large block transfers from one area of the disc to another. Incoming problems share the 235 in short bursts of time on a rotating basis.

When the system was first put into operation, only one algebraic compiler, BASIC, was available. Subsequent additions have included: TEACH, a system which allows an instructor to code BASIC programs to analyze the results of a student program while the student's program is running; a fairly complete version of ALGOL 60; a machine-language interpretive program called DIP; and a program maintenance system called EDIT. Lochner (65-903) covers the approach used to develop the system and response by users.

Berkeley (65-904) demonstrates the use of a remote console of the Dartmouth system situated in a mathematics classroom at Phillips Exeter Academy. The student-contrived programs for Tic Tac Toe and Roulette are in BASIC.

5. CLASS

The Computer-based Laboratory for Automated School System (CLASS), developed by the System Development Corporation, provides instruction for up to 20 students under the control of a computer. In addition to a laboratory-type course in statistical inference, CLASS has been used for computer-based student counseling and field evaluation of an elementary Spanish course (65-402, 64-403).

6. System 473L

System 473L is a command-control system operated by the U.S. Air Force. It includes an automated instructional subsystem which teaches AF console operators the query language to be used in communicating with main system (64-1203).

7. Edison Responsive Environment

The Edison Responsive Environment is a jam-proof electric typewriter, slide projector, and tape recorder, all connected to a self-contained logic network or computer. Currently the logic is not time-shared among students, although it may be in the future. The system is used primarily to teach pre-school children language arts (65-502).

8. SAKI

The best known of the machines developed in the UK by Gordon Pask is SAKI, which instructs the student in keyboard operation (65-103). Errors and response times were calculated by SAKI, compared with standard performance and fed back to the student. It was not a time-shared system.

9. COBIS

The Computer-Based Instructional System (COBIS), located at the Electronic System Division, Hanscom Air Force Base, Bedford, Mass., is based on a DEC PPD-1(65-301). It has three principal features: (1) A light pencil is used as the medium of communication between the student and the computer. (2) The student indicates his degree of certainty for each alternative in a multiple choice array by adjusting the bars of light next to each answer on a CRT screen. (3) The computer considers both the student's answers and his degree of certainty when branching to remedial sequences or further steps. A special scoring system has been developed for this purpose.

10. The Socratic System

In the Socratic System, developed by Bolt, Beranek and Newman, Inc., around the DEC PDP-1 computer, the teacher and student carry on a dialogue in depth (65-1001, 65-703, 65-302, 64-601, 64-503). The computer states a problem, sets up conditions, asks questions, provides requested data, and answers questions, while observing the student's course of action in a task. The system has been applied to instruction in medical diagnosis and business decision making.

11. SOCRATES

SOCRATES is a time-shared system developed at the Training Research Laboratory of the University of Illinois. It is adaptive in three ways: (1) it learns about the student as it teaches him, (2) it may make decisions about the effectiveness of the rules used to teach the student, and (3) it may make decisions about the criteria which are used for evaluating performance.

The first SOCRATES student interface was put on-line in May 1964. SOCRATES uses an IBM RAMAC external disc memory of 2 million-digit capacity to supplement internal storage in an IBM 1620 computer. The system may display any of 1500 frames of film to the student. An IBM 1800 central computer and a Westinghouse sound box with 1000 stored one-sec. messages under computer control are on the way (65-701, 64-1101, 64-901, 64-602).

12. PLATO III

PLATO III is the most recent in a series of systems devised by the Coordinated Science Laboratory at the University of Illinois. The central computer, a CDC 1604, is connected to 20 student stations. The system uses an "electronic book and blackboard" video display system on which are superimposed both optical and CRT displays. The basic logic is tutorial and intrinsic. The PLATO compiler (CATO) is compatible with FORTRAN.

One capability of PLATO is Responsive Environment Programmed Laboratory (REPLAB). A motion picture film is first shown using auxiliary equipment, then PLATO presents questions. Two brief programs have been used, one in physics (6th grade) and one in clinical methods for student nurses. A mathematics laboratory called PROOF is also in use (64-1001, 64-401, 62-901, 62-802, 62-702, 62-401).

D. STUDENT INTERFACE EQUIPMENT

The student interface is the junction, or communication link, between the student and the computer. It is an essential part of every CAI system. All systems in the literature reviewed are based upon two-way communication -- indeed, obtaining such an interaction or "conversation" between pupil and computer is a major reason for using computers as teaching devices.

1. Display Techniques

The problem is twofold: the computer's output must be displayed to the student and the student's output (responses) must be fed into the computer. A simple way to achieve both functions is to use an electric typewriter. This was the technique used in the early work in the IBM laboratories (64-302, 59-001), and is an alternative with the present IBM "Coursewriter" system (65-803, 64-001). A typewriter (teletype) keyboard is also used for CAI with the Dartmouth Time-Sharing System (65-903), and with the Bolt Beranek and Newman "Socratic System" (65-1001, 64-601).

While a typewriter has the advantage of being an extremely flexible student response devise, its relatively slow speed limits its desirability as a display for the student. This limitation has been circumvented by using a visual display with more rapid access to computer storage (e.g. a CRT), by using an auditory display, and by storing messages on film or tape and displaying these directly. In the latter case, the computer is used only to determine the location of the message; therefore requirements for storage capacity are greatly reduced.

In IBM's "in house" research on their CAI system, CRT displays are used in an experimental employee training course in two locations (Washington, D. C. and San Francisco), using the IBM 1050 input-output station (65-803). Other training locations are using the typewriter alone. This system uses telephone data lines for communication with a central computer, and has additional telephone lines for direct voice communication with a human instructor. The SOCRATES system at the University of Illinois Training Research Laboratory is scheduled to have a CRT display in the near future (64-1101), and COBLS at the Air Force's Decision Science Laboratory uses a CRT display exclusively (65-501, 65-301).

PLATO, developed at the Coordinated Science Laboratory, University of Illinois, uses a unique combination of techniques (64-702, 64-401). Basically a video display, it is described as an "electronic book and blackboard." The "book" section allows individual microfilm slides to be displayed on the television screen, while the students' answers to questions, etc. are presented on the same screen in a manner analogous to writing on a blackboard.

In its present configuration, SOCRATES employs a microfilm display (65-701, 65-101, 64-602). The central computer (an IBM 1620) controls a random access microfilm transport which displays one of 1500 instructional slides to the student. Microfilm slides are also used in the computer controlled economics games developed by the Board of Cooperative Educational Services of Northern Westchester County, New York (65-203). While the basic display is a typewriter (IBM 1050), a random access slide projector provides auxiliary information to the student.

A quite different display was utilized by Teleregister for United Air Lines which makes collateral use of their "Instamatic" reservation system for training ticket agents (64-404). The instructional materials are presented in book form as multiple-choice questions. A regular reservation printout card is used to display answers.

A combination typewriter-voice-slide projector display is used in the Edison Responsive Environment or "talking typewriter" (65-502). The typewriter both types and names letters and words when its keys are properly depressed. It also displays slides of letters or words as cues to the response of typing them.

2. Response Techniques

A typewriter keyboard is the basic response device in the IBM Coursewriter system (65-803), the Board of Cooperative Educational Services economics games (65-203), the Dartmouth Time-Sharing System (65-903), and the Edison Responsive Environment (65-502). The PLATO system also uses a typewriter (teletype) keyboard, but in its various configurations may assign many different purposes to the keys (64-702, 62-401). Plastic caps are placed over special purpose keys to identify them.

Two syst   use keyboards other than that of a standard typewriter. SOCRA   has fifteen buttons which may be asigned a variety of functions depending on the program -- e.g. letters, digits, yes-no, etc. (65-701, 65-101, 64-602). The Teleregister-United Air Lines system uses the "Instamatic" agent-set keys with an overlay to reidentify them for training (64-404).

The student responses to CCBIS are made with a light pencil (65-501, 65-301). A bar of light is displayed next to each multiple-choice answer on a CRT. These bars represent the probability of correctness of each answer, and the student indicates his degree of certainty for each answer by adjusting the length of the bars with his light pencil. Both the answer and the degree of certainty are considered by the computer when branching or advancing to the next level.

Two publications are available which consider general interface problems. The first, by Randa of MIT (62-201), is concerned with basic hardware. The second, by Glaser, Ramage and Lipson (65-1204), deals with more specific instructional problems and considers the choice of equipment for different subject matter, for special education, and for research.

III. LANGUAGES

CAI languages can be divided into programmer-oriented and user-oriented languages. The users in turn fall into two classes: teachers and students. Teacher-oriented languages, such as COURSEWRITER, are the medium by which the instructor tells the computer what assertions to make to the student, especially as a function of the student's previous response(s). A student language is the medium by which the student interacts with the computer. When the student is to use the computer as a tool or laboratory to solve a problem or a sequence of problems, the student language may be a compiler (e.g. BASIC) or a problem-oriented or simulation language (e.g. SEPOL).

A. STUDENT LANGUAGES

1. BASIC. A programmer-oriented algebraic compiler available for the Dartmouth time-sharing system (65-903). Has been used by secondary school students to program games, "Tic-Tac-Toe" and "Roulette," on the Dartmouth system (65-904).

2. TEACH. A system which allows an instructor to analyze the results of a student's program while the student's program is running. Used in the Dartmouth time-sharing system (65-903).

3. SIMIN (Simulated Statistical Inference Program). Used to teach inferential statistics at Michigan State University; 66 types of computer output are available to the student (64-002).

4. Natural Language (e.g. English). Most query languages take the form of artificial, quasi-English languages which bear slight resemblance to ordinary English, but are sufficiently simple syntactically and sufficiently mnemonic lexically that user can master them with less difficulty than more specialized languages. Keyser (63-902) and Raphael (64-604) outline linguistic theories which increase the possibility of the use of natural English as a query language in CAI systems.

5. Graphic Language. Although the ideal is generally held to be a natural language, e.g. English, Duggar et al (65-102, 63-1202) cite the potential advantages of graphical symbology and logic flow diagrams over English as a language for teaching logic and decision tasks to students.

B. TEACHER LANGUAGES

1. COURSEWRITER. Presented by IBM as an aid in writing and presenting instructional material through the IBM 1401 and 1410 computers and 1050 Data Communications System. Course preparation requires no special knowledge of computers. COURSEWRITER has a dozen basic instructions and its use can be learned in one or two hours (64-001). COURSEWRITER has been used to program statistics (64-302) and several other subjects.

2. MENTOR. A teacher-oriented language and compiler for the Socratic System in which language conventions have been chosen so as to make the intention of the control statements comprehensible to the teacher as well as the programmer. Thus, the teacher may "converse with the computer when drawing up his instructional program in much the same way that the student talks with the machine during a lesson (64-804, 64-803)."

3. CATO. The PLATO compiler is compatible with FORTRAN, facilitating its use by non-technical persons who wish to prepare lessons for PLATO (64-1001).

C. PROBLEM-ORIENTED STUDENT LANGUAGES

1. STRESS (STructural Engineering Problem Solver). Used by the graduate class in structures in the MIT Civil Engineering Dept. to design complete structures (64-301).

2. COGO (COordinated GeOmetry). Another problem-oriented language used at MIT. Includes 70 special words including "point," "distance," "locate," etc. (64-301).

3. SEPOL (Soil Engineering Problem-Oriented Language). Used in MIT Civil Engineering Dept. (64-301).

D. COMPILERS

1. ALGOL 60. A programmer-oriented compiler available on the Dartmouth time-sharing system (65-903).

2. DIP. A machine-language interpretive program used on the Dartmouth time-sharing system (65-903).

3. EDIT. A program maintenance system available on the Dartmouth time-sharing system (65-903).

IV. INSTRUCTIONAL THEORY

Some of the systems described in Chapter II are the result of the systematic consideration by behavioral scientists of the facts and theories of learning and instruction. Such systems are "custom-made" for instruction. This is particularly true of the tutorial systems.

Other systems, particularly the laboratory systems, simply represent the application of the computer as a problem-solving tool to didactic problem-solving as a technique of instruction.

Whether guided by behavior theory or the seat of their pants, each decision by a system designer or a programmer contributing to the design or implementation of a system is a contribution to a miniature theory of instruction. The system diagrams, flow charts and programs of CAI are theory fragments, or "logics," based on some assumptions about the learning process.

At the same time, data collected in computer-directed learning exercises are being added to the store of information on the instruction and learning processes. In some systems, such as SOCRATES, a fair amount of time is scheduled for systematic computer-controlled studies of learning and instruction variables. In other systems, observations on learning effectiveness are informal and incidental to application.

In reviewing the studies of learning and instruction related to CAI, we have separated the reports into three categories: (1) Experimental studies of learning, such as the learning of paired associates, which seem to have a direct application to CAI. (2) Analysis of the instructional logics on which various systems are based. (3) Optimization of learning by the design and evaluation of instructional sequences (programming).

A. STUDIES OF LEARNING

1. Paired Associates Learning

Dear et al (65-304) compared a learning-optimization procedure derived from a stimulus-sampling model and another simple presentation strategy in training subjects in several paired-associates tasks. The "optimal" strategy was found to be no better than the second procedure. It appeared some representation of short-term memory must be incorporated in the model. The experiments were conducted in the SDC CLASS system.

Izawa and Estes (65-802) found that the learning of paired-associate items was affected to a major extent by the ratio of reinforced (R) to test (T) trials and by their arrangement in various repetitive sequences. For example, overall the sequence RTRT... appears more nearly optimal than RRTRRT.... A stimulus-fluctuation model provides a somewhat better account of the acquisition and retention phenomena than a one-element model.

2. Learning Set

Using conventional programmed textbook formats, Reynolds, Glaser, and Abma (64-1101) found learning sets were formed by all learners using programmed instruction, regardless of individual differences. PSI research at the University of Illinois Training Research Laboratory includes an effort to establish whether a learning pattern in programmed instruction is one of proactive inhibition or learning set (64-602).

3. Inspection Behavior

Rothkopf (63-103) proposes a model for learning from self-instruction programs consisting chiefly of written sentences. The model suggests that inspection behavior plays a critical role in the effectiveness of self-instruction programs.

4. Computer Simulation of Human Learning and Problem-Solving

Human learning and problem-solving processes have been studied by computer simulation (64-505). Evans (64-1103) has devised a computer program (in LISP) which solves by heuristic methods a wide variety of intelligence test problems of the geometric analogy type, viz: "Figure A is to Figure B as Figure C is to which of the following?"

B. INSTRUCTIONAL LOGICS

Conversational interaction as simulated on the computer can be classified in two major categories: tutorial and Socratic. In the tutorial model, information is asserted to the student and he is required to respond to a question. There are three kinds of tutorial programs: linear, intrinsic, and adaptive. Following Skinner, in a linear program the pace of instruction is changed to fit the student, and stimuli to which the student responds incorrectly may be repeated. Otherwise the linear program treats all students alike.

In an intrinsic program, usually made up of a sequence of multiple choice items, the selection of each assertion is based on the student's response to the previous stimulus. Thus, by branching, each student may take a different path through the same subject-matter. The program is adaptive in pace and in the amount of instruction each student receives.

In an adaptive tutorial program, each assertion is based on an extensive history of student responses to the program. Stolurow (65-702, 65-701, 64-602) advocates an adaptive system in which each stimulus is based, not only on the student's responses to the program, but on his pretutorial characteristics of aptitude and personality. He calls this an ideomorphic system and has illustrated in operational terms (64-901) the way in which different characteristics of learners are used in ideomorphic programming (implemented as a two-stage decision process).

So much for the tutorial model. The second major interaction model is the Socratic system. As described by Feurzeig etal (65-1001, 65-703, 65-302, 64-601, 64-305), the Socratic system goes beyond the tutorial programs by allowing the student to make assertions or ask questions of the computer. The computer bases its response to a question or assertion on its history of the interaction to that point.

In the Socratic system the student is to (1) deduce relationships among elements of a solution, i.e. form concepts; (2) ask questions and make assertions and always obtain a meaningful response; and (3) use natural language. The computer control language should (4) employ conditional expressions taking into consideration the history of the learner's actions, and (5) the compiler should be easy for the teacher to use (65-302). The Socratic system is also, therefore, adaptive.

Most of the systems described in this report fit into one of the foregoing categories, although one may overlap another. The basic logic in PLATO II is tutorial and intrinsic, but the student can control the presentation with special function keys including "reverse" which permits review of slides, "help," and "aha" which permits termination of a "help" sequence (62-401).

PLATO III is a tutorial program like PLATO II, but it provides up to eight help or error correction sequences instead of one. "Evaluators" can also be inserted. These compare student performance with specified criteria and branch to additional expository materials until the criteria are met. The program is thus made more adaptive (64-707). In a PLATO III Inquiry Training Program a motion picture is first shown using auxiliary equipment. Then PLATO presents questions. The student may try to answer, or request additional information (64-401).

COURSEWRITER is also tutorial and intrinsic (64-001). An early realization of an adaptive logic in a teaching machine was brought off by Pask in SAKI (65-103).

C. SIMULATION

Not all CAI is as directive as a tutorial program, or even as the Socratic system. In some applications the computer creates a simulated environment and provides an opportunity for the student to interact with that environment, learning by trial and error.

For example, the computer may simulate a laboratory in which the student is able to initiate the same operations he would in a real laboratory. SDC has created a simulated laboratory for statistics (65-402, 64-002). The MIT Dept. of Civil Engineering has developed structures and stress problems which undergraduate and graduate students solve on the computer (64-301). In the MIT applications, the computer not only creates the simulated environment, but is used as a tool in the solution of the problem, in much the same way it will be used to solve a real engineering problem.

IBM and the Board of Cooperative Educational Services (BOCES) in Northern Westchester County have developed a simulated environment for teaching economics to elementary school students (65-201, 64-1202). The purpose of this simulation is to develop in the student (1) an understanding of the processes at work in a developing civilization, and (2) the ability to make decisions regarding multifactored situations, taking account of several conflicting values or goals. The student assumes the role of a priest-king of a Sumerian city-state in 3500 B.C. Through a remote typewriter terminal, the computer (IBM 7090) gives him economic conditions and asks for decisions (e.g. How much of this year's grain harvest should be set aside for next year's planting?). BOCES has a second simulation, the Sierre Leone Development Game, in development (65-203).

The Socratic System also creates a simulated environment (65-1001, 64-804, 64-601). For example, for instruction in medical diagnosis, a medical patient and hospital facilities are simulated. In the management exercise, a business is simulated. Unlike the MIT engineering exercises or SOCRATES economic games, the Socratic system includes evaluation of the [illegible] performance in the feedback, reducing the trial-and-error character of the instruction. This feature is available in the real environment when an instructor is present.

In summary, CAI programs may be arranged on a continuum extending from the highly directive programs, such as COURSE-WRITER, at one end, to the non-directive at the other, such as the MIT civil engineering problems. The Socratic System, the Sumerian Game, and SIMIN lie somewhere in between these two extremes.

V. PROGRAMMING

A. SEQUENCE VARIABLES

In structuring a program, even a linear program, sequence variables fall into at least three classes related to (1) the nature of the content, (2) the nature of the response, and (3) the presence or absence of reinforcement. The programmer must first decide in what order to present subconcepts to be learned. Then he must decide whether these will be presented as rules, examples, or both. Finally, he must determine the sequence of reinforced and non-reinforced (test) items, including the repetition of reinforced items as review frames. Hickey and Newton (64-103) have discriminated between these variables in a study of the structure of the knowledge space and the relationship between a multi-dimensional knowledge space and the one-dimensional teaching sequence.

In experimental studies of sequence variables, some investigators have found differences in learning effectiveness, others have not. Stolurow and his colleagues (64-701) found no difference in learning effectiveness as a result of prior experience with three different components of a subsequently learned mathematics task. Nor did the order in which items appeared in a program seem to make a difference in learning when one sequence is considered orderly and the other random. Newton and Hickey (64-103) report some effects, however. Using 12 versions of a 59-frame program teaching a concept in economics, they varied the order and position of subconcepts and the direction of the program with respect to definitions. The sequence variables did not affect error rate, but some did affect time to complete the program and achievement.

Sheridan and Mayer (65-305, 63-1202) and Bio-Dynamics, Inc., (65-403) describe a phylogenic strategy for sequencing the content of a program that teaches the function and structure of a system. Following the phylogenic sequence, the trainee behaves as an inventor, causing the subject system to proceed in its development through the various stages, or "phyla," in its evolution, either hypothetical or real.

Content sequence was investigated by Dear et al (65-304) in paired associates learning experiments. One optimization theory for the presentation of stimulus items was based on a stimulus-sampling model of a learning experiment. However, this strategy was not found superior to another simple presentation strategy when the two were compared experimentally. It appeared some representation of short-term memory must be incorporated in the model.

Still on the topic of content sequence, Stolurow (64-002) thought it likely that given programmed instruction sequences generate learning patterns corresponding to traditional concepts of learning theory. In one study he sought to determine whether an undesirable learning pattern generated by a program was one of proactive inhibition or learning set.

In a study of repetition and review in programmed instruction, Reynolds et al (64-1204) found that repetition above the usual level in a linear program is of no benefit; but that spaced review is potentially beneficial.

Two of the principal elements in programmed instruction sequence are reinforced (R) trials or frames, and non-reinforced or test (T) trials. Izawa and Estes (65-302) studied the optimal sequencing of R and T trials in teaching paired associates. The course of learning was affected by the ratio of R's to T's and by their arrangement in repetitive sequences. Overall the sequence RTRT... appears to be most nearly optimal. They conclude that a stimulus-fluctuation model gives a better account of acquisition and retention phenomena than a one-element model.

Hershberger (65-205) compared three intervals of delay between the initial stimulus presentation and subsequent testing. In the first sequence, S reread the stimulus item <u>before</u> the test frame; in the second, <u>after</u> the test frame; and in the <u>third</u> only the correct answer was exposed after S responded. There was an effect on errors in the program, but not on achievement. In any case, however, self-testing was superior to no self-testing.

B. DECISION STRUCTURE

1. Instructional Alternatives

Linear programs give all students the same instructional material. To accommodate individual differences, intrinsic, branching or adaptive programs may make available alternative materials, basing the selection of material on the student's most recent response or on a more extensive response history. For example, to instruct operators of an Air Force information retrieval system in the query language used in the system, Clapp et al (64-1203) used a computer-directed program that followed a modified "main trunk" branching logic. The "main trunk" was the linear program including pretest, instructional sequence, and posttest.

The difference between alternative stimulus materials may simply be in the amount of cueing or prompting, the amount of information in frames (step size), repetition of regular frames, and presentation of special remedial frames. Glaser and his associates (64-1002) did not demonstrate an increase in learning efficiency for the first two methods over a straight linear program. Kress and Cropper (65-901) also varied the amount of prompting in a program presented at a single fixed pace. Adjusting the amount of prompting was not as effective as subdividing the population into groups homogeneous with respect to their pacing requirements. Stolurow et al (64-701) found that, although S spent less time on a large-step program, there was no significant difference between performance on large and small-step programs in terms of achievement.

Using programmed texts, Kaufman (63-102) varied the amount of remedial material available to the student in an intrinsic program. He did not find a significant difference among the three amounts with respect to time to complete the program or achievement.

2. Decision Logic

Intrinsic and adaptive programs base the selection of alternative material on the student's most recent response, or on a more extensive response history. Several logics for the selection of materials at each step in a program have been proposed. In a computer-assisted study of speed reading, Strollo (64-504) based the choice of instructional material on a statistical evaluation of the student's total behavior in comparison with other students' total behaviors, the statistics changing as new students took the course.

Shuford (65-501) and Baker (66-301) have described a decision structure by which the computer considers both the student's answers and his degree of certainty when branching to remedial sequences or further steps. A special scoring system has been developed for this purpose. No "errors" are made by the student. He progresses through levels of certainty and the computer always acts in the direction of raising his certainty.

C. FRAME CONSTRUCTION

Do affective verbal terms undermine the effect of formal, logical, or quantitative terms that appear in programmed instruction frames? Frase (65-908) studied this question in the learning of syllogisms. He found that, in judging the validity of syllogisms in which the affective terms were highly inconsistent with the formal terms, the Ss at first made snap judgements that were incorrect, but later were more deliberate. A moderate level of incompatibility resulted in the greatest number of

Stolurow and his associates (64-761) report no consistent difference in learning effectiveness as a result of overt and covert responding to programmed instruction material in mathematics. In comparing the effects of presenting programmed instruction material before and after a teacher's lecture, however, they did find it more effective to present the programmed instruction first.

Rothkopf (63-193) describes a conceptual model for learning from self-instruction programs that consist chiefly of written sentences, stressing temporal contiguity as an important association-forming principle. The model proposes that inspection behavior plays a critical role in determining the effectiveness of self-instruction programs.

D. PROGRAM GENERATION

Clapp and her colleagues (64-1203) describe the development of the computer-based instructional materials used to train operators of the 473L system in the query language. Emphasis is on the training strategy, the operational specifications and the flow chart and procedural flow diagrams.

Swets and Feurzeig (65-1001) report that 30 hours were required for a physician and computer programmer to prepare the medical case used to demonstrate the application of the Socratic System to instruction in medical diagnosis. An additional 30 hours of computer programming and clerical transposition were needed, plus several hours for editing the English prose. The student's vocabulary consists of 40 questions and 35 declarative statements.

Klaus (65-909) believes one hour of instruction per week, or one week per year, is a reasonable production standard for the development of programmed instruction materials. He concludes that, unless programming itself is automated, it will often be necessary to resort to conventional methods of instruction. He describes a seven-step procedure he has followed for gradually automating instruction which he believes can save substantial amounts of programming effort.

Stolurow describes the use of SOCRATES to develop a technology of generating learning materials (65-701). The procedure is based on a set of codes for instruction frames which allow the computer to judge, from a student's history of frames, which frames are relevant for his future use (64-602).

In the developmental evaluation of programmed instruction materials, how many students should be used as a basis for accepting or rejecting frames? Francke et al (55-204) related this question to the problem of the statistician in testing a large number of hypotheses. The hazards of small samples (N=5) were examined. Wide variations in efficiency among samples of a given size were observed in terms of (a) rejection of acceptable frames, and (b) retention of unacceptable ones. It was recommended that samples be as large as practical.

VI. EVALUATION

The ultimate criterion for automated instruction is student achievement per unit time and cost, but there are many intermediate or related criteria, such as error rate, reaction time, data rate requirement placed on the system, student reaction, etc. Eight performance criteria are discussed in connection with the evaluation of PLATO II (62-702).

Several investigators have compared the learning effectiveness of CAI and more conventional instruction. At the University of Illinois, nine students in an introductory course in digital computers took three lessons via PLATO II. Their examination grades parelleled those of the rest of the class and they were enthusiastic about CAI (62-702).

Grubb and Selfridge (64-302) have described a typical experimental design used to compare CAI performance with instruction via programmed text or lecture. The three performance criteria were mean instruction time, mean review time, and average achievement score. The course was the first half of a one-semester course in psychological statistics. Students under CAI covered the material in 5.3 hrs. compared with 49 hrs. for the lecture mode and 12.2 hrs. for the programmed text. The average achievement score in the CAI mode was 94.3 compared with 58.4 in the lecture mode.

Student reactions to CAI were reported by Wodtke and his associates (65-902) who found that some students, especially low achievers, considered the CAI presentation on a remote terminal of the IBM Yorktown system too rapid. They suggest student-controlled pauses.

VII. ADMINISTRATION

Based on PLATO system experience, Stolurow (64-402) believes CAI may be made available for as little as 10 cents per student hour.

Grubb (65-906) points to the economic advantage of assigning two students instead of one to each CAI terminal. He investigated the pairing of students in a CAI course in statistics. His results indicate that students paired on the basis of their College Entrance Examination Board verbal scores will do as well on the final examination as students who study individually at the CAI terminals.

APPENDIX A

CAI Subject Matter

Subject	Call No.	Serial No.
Mathematics and Statistics		
Algebra, UICSM	222/F897	64-204
Arithmetic, binary	101/R234	59-001
Mathematics: Algebraic proof	424/C778D	64-1001
	424/C778C	64-702
Number squaring and cubing	730/M266	65-910
Statistics	730/M266	65-910
	730/G885	65-906
	644/G885	64-302
Statistics, inferential	022/F948	64-002
	116/C324	65-402
History		
History, American	730/M266	65-910
Languages		
English	730/M266	65-910
Logic: Syllogisms	400/F841	65-908
Spanish	116/C324	65-402
Science		
Medical diagnosis	435/F426	64-601
	434/S975	65-1001
	433/F426	64-804
Physics: Bi-metal strip expt.	424/C778D	64-1001
	424/C778C	64-702
Science, elementary	444/W769	65-202
Economics		
Economics	448/M739	65-201
	424/C778	64-1202
Business and Accounting		
Accounting	430/F426	64-503
Engineering		
Structures	746/M414	64-301

Subject	Call No.	Serial No.
Computer Technology		
Computer approaches to engineering problems	746/M414	64-301
Computer programming (473L)	644/C589	64-1203
Computer use	644/M468	65-801
Computers, digital	424/C778A	62-702
Games		
Bridge	730/M266	65-910
Roulette	731/B512	65-904
Tic-Tac-Roe	731/B512	65-904
Reading and Spelling		
Reading	730/M266	65-910
	640/P651	65-502
	411/S919	64-504
Spelling	730/M266	65-910
Skills		
Keyboard operation	620/E97	65-103
Metal surfaces, smoothness of	420/M191	63-901
Nonverbal sounds	328/S975	62-601
Nursing, clinical methods in	424/C778B	64-401
Task logic	118/S552	63-1202
Typing	640/P651	65-502

APPENDIX B

INSTITUTIONS WITH CAI

Institution	System
Stanford Univ. (Stan.)	Stanford I & II
System Development Corp. (SDC)	CLASS
Carnegie Inst. of Technology	
Electronic Systems Div., Air Force (ESD)	COBIS
Bolt, Beranek and Newman (BBN)	Socratic System
Univ. of Illinois, Training Research Lab (TRL)	SOCRATES
Univ. of Illinois, Coordinated Science Lab (CSL)	PLATO
Mass. Inst. of Tech., Civil Eng'g. Dept. (MIT/CE)	
Mass. Inst. of Tech., Operations Rsch.Grp.(MIT/OR)	
Mass. Inst. of Tech., Project MAC (MIT/MAC)	MAC
Mass. Inst. of Tech., (MIT/ESL)	
IBM Watson Rsch. Lab. (IBM/York.)	IBM Yorktown
IBM Poughkeepsie	IBM Poughkeepsie
IBM Advanced Systems Div. (IBM/Los Gatos)	
Univ. of Pittsburgh, Learning Research and Development Center (Pitts/LRDC)	
United Air Lines (UAL)	Instamatic
Brooks Foundation	
Bell Tel. Labs (BTL)	
Penn. State Univ. (PSU)	IBM Yorktown
Florida State Univ. (FSU)	IBM Yorktown
Harvard Univ. Computation Center (Harv.)	TACT
Univ. Calif., Santa Barbara (UCSB)	
Univ. Calif., Los Angeles (UCLA)	
Univ. Calif., Irvine (UCI)	
Univ. of Omaha (UO)	
McGraw-Edison	ERE
Board of Coop. Educational Services (BOCES) of North Westchester County	

APPENDIX C

CENTRAL COMPUTERS

1. GE-235 & D-30. Central equipment in the Dartmouth system. The D-30 is the master computer and schedules all nontrivial operations in the 235. The D-30 can communicate with the 235 or the dual access file and handles I/O from and to the teletypes, allocates and releases disc space and analyzes and responds to user commands. The 235 is used for editing, compilation, program runs, and large block transfers from one area of the disc to another. Incoming problems share the 235 in short bursts of time on a rotating basis (65-903).

2. IBM 1440. Central computer in the IBM system at the IBM Watson Labs, Yorktown, N. Y. Also connected to Penn State Univ., Florida State Univ. (65-902, 65-803, 64-001).

   Central computer in the IBM time-shared system at Poughkeepsie, N.Y., with student stations in Philadelphia, Los Angeles, San Francisco, and Washington, D. C. (65-803).

3. IBM 7094. Two 94's located in the Computation Center and Project MAC at MIT are connected to typewriters in Rm 1-150 the Civil Engineering Dept. (64-301).

4. IBM 1620. Rm 1-150 in the MIT Civil Engineering Dept. contains an IBM 1620 (360,000 bit memory plus 32,000,000 bit disc pack) (63-301). SOCRATES has been developed on a 1620 with 2 million digit RAMAC disc memory at the Univ. of Illinois Training Research Lab. (64-602).

5. IBM 650. Before changing over to the present 1410-1440, the IBM Yorktown system was based on an IBM 650 RAMAC with disc file for 6 million digits and 0.8 max. access time (64-302, 59-001).

6. IBM 1401 or 1410. The Air Force Electronic Systems Division used the System 473L computer (IBM 1401 or 1410) to teach console operators the query language used by the operators to retrieve information from the 473L system (64-1203).

7. IBM 7090. The Board of Coop. Educational Services (BOCES) of Northern Westchester County used an IBM 7090 time-sharing system to present computer-based economics games to sixth-grade students (65-203, 65-202).

8. DEC PDP-1. Central computer in the BBN Socratic System at Bolt Beranek and Newman (64-601, 62-601) and in the Stanford Univ. system (65-001).

9. CDC 1604. The central computer for PLATO III (64-1202, 64-1001, 64-401).

10. ILLIAC. The central computer for PLATO I and II. (64-1202, 62-401).

11. Bendix G-20. Installed in Carnegie Institute of Technology Computation Center for use as a teaching machine for students in mastering certain drill problems in math., engineering, and physical sciences (61-701).

12. UMShN. Central computer in the Radon complex at the Academy of Sciences of the Ukraine (64-201).

13. Ural-1. Central computer at the Kiev Higher Radiotechnical Engineering School (64-201).

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* ENTELEK Incorporated 42 Pleasant Street Newburyport, Mass. 01950	2a. REPORT SECURITY CLASSIFICATION 2b. GROUP

3. REPORT TITLE

COMPUTER-ASSISTED INSTRUCTION: A REVIEW OF THE LITERATURE - JANUARY 1966

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Semiannual report (July 1 - Dec. 31, 1965)

5. AUTHOR(S) *(Last name, first name, initial)*
Hickey, Albert E.
Newton, John M.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
January 1966	44	100

8a. CONTRACT OR GRANT NO. Nonr-4757(00)	9a. ORIGINATOR'S REPORT NUMBER(S) 66-101
b. PROJECT NO. NR 154-254	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. AVAILABILITY/LIMITATION NOTICES

Qualified requesters may obtain copies of this report from DDC.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Personnel & Training Branch (Code 458) Office of Naval Research Washington, D. C. 20360

13. ABSTRACT

A selective review of 100 documents related to computer-assisted instruction (CAI). Headings: Applications, Systems, Languages, Instructional theory, Programming, Evaluation and Administration. Appendices list programs by subject matter, institutions with CAI, and central computers in CAI systems. Review will be updated semiannually.

DD FORM 1473

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Literature survey						
Computer-assisted instruction						
Programmed instruction						
Learning theory						
Theory of instruction						
Time-sharing						
Simulation						
Systems analysis						
Man-machine interface						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through
_______________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through
_______________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through
_______________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, *or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.